

Amendments to the Claims:

The following listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1 Claim 1 (currently amended): A method of maintaining ~~http~~ session data in a server system
2 serving a network, said server system including at least one network server and at least one
3 database server, said method comprising the steps of:

4 (1) storing in a memory local to said first network server, session data for a plurality
5 of sessions serviced by said at least one network server;

6 (1) (2) storing in a database local to said database server, copies of said session data
7 for a plurality of sessions serviced by said at least one server;

8 (2) (3) performing contemporaneous time out testing of particular session data stored
9 in said memory local to said first network server for one of said plurality of sessions every
10 time a request is received for said particular session data prior to utilizing said particular
11 session data, and not invalidating a copy of said particular session data in said database even
12 if said contemporaneous testing has indicated that the corresponding session has timed out;
13 and

14 (3) (4) performing an invalidation procedure on said copies of said session data in said
15 database at a particular time that is independent of when said contemporaneous testing is
16 performed.

1 Claim 2 (original): The method of claim 1 wherein said session data comprises an
2 HttpSession object of a Java servlet application program interface (API).

1 Claim 3 (original): The method of claim 2 wherein said Java servlet APIs are J2EE servlet
2 APIs.

1 Claim 4 (currently amended): The method of claim 2 wherein step (1) comprises the step of:
2 (1.1) creating ~~[[an]]~~ said HttpSession object for a session upon initiation of said
3 session.

1 Claim 5 (original): The method of claim 4 wherein step (1) further comprises the step of:
2 (1.2) updating said HttpSession object for said sessions as said session progresses.

1 Claim 6 (original): The method of claim 5 wherein said server system comprises a plurality
2 of Java Virtual Machines (JVMs) of which different ones of said JVMs may service different
3 http requests corresponding to a single http session and wherein said database is accessible to
4 each of said JVMs.

1 Claim 7 (currently amended): The method of claim 6 wherein step (1) further comprises the
2 step of:

3 (1.3) storing said HttpSession object for each session handled by a JVM in a memory
4 local to a network server running said JVM;

5 and step (2) further comprises the step of:

6 ~~(1.4)~~ (2.1) writing a copy of said HttpSession object for each session stored in said
7 ~~local memories to said database~~ local to said database server.

1 Claim 8 (original): The method of claim 7 wherein said plurality of JVMs run on a plurality
2 of network servers.

1 Claim 9 (original): The method of claim 8 wherein said server system services the World
2 Wide Web.

1 Claim 10 (original): The method of claim 1 wherein said particular time is a function of a
2 periodic interval.

1 Claim 11 (original): The method of claim 10 wherein said periodic interval is a day and said
2 particular time is a time of day.

1 Claim 12 (original): The method of claim 11 wherein said time of day is a time of day that a
2 load on said database is expected to be low.

1 Claim 13 (currently amended): The method of claim 1 further comprising the steps of:
2 ~~[[{4}]]~~ (5) periodically determining a load on said database; and
3 wherein said particular time is a function of said determined load and a predetermined
4 interval.

1 Claim 14 (currently amended) The method of claim 1 wherein said invalidation procedure
2 comprises invalidating all of said ~~sessions~~ copies of said session data stored in said database.

1 Claim 15 (currently amended) The method of claim 1 wherein said invalidation procedure
2 comprises the steps of:

3 ~~(3.1)~~ (4.1) for each copy of said session data in said database, determining if said said
4 session has timed out;

5 (3.2) for each session that has timed out, invalidating the corresponding copy of
6 session data in said database.

1 Claim 16 (currently amended): A server system serving a network comprising:

2 at least one network server having a local memory;

3 ~~a memory;~~

4 at least one database server having a database;

5 a first computer program adapted to store in said memory local to said network server
6 session data for a plurality of sessions serviced by said at least one server;

7 a second computer program adapted to store in said database copies of said session
8 data;

9 a ~~second~~ third computer program adapted to perform contemporaneous time out
10 testing of particular session data for one of said plurality of sessions every time a request is
11 received for said particular session data prior to utilizing said particular session data, and
12 further adapted to not invalidating to allow a copy of said particular session data in said
13 database even if said contemporaneous testing has indicated that the corresponding session
14 has timed out; and

15 a ~~third~~ forth computer program adapted to perform an invalidation procedure on said
16 copies of said session data in said database at a particular time that is independent of when
17 said contemporaneous testing is performed.

1 Claim 17 (original): The system of claim 16 wherein said session data comprises an
2 HttpSession object of a Java servlet application program interface (API).

1 Claim 18 (original): The system of claim 17 wherein said Java servlet APIs are J2EE servlet
2 APIs.

1 Claim 19 (currently amended): The system of claim 17 wherein said first program creates
2 [[an]] said HttpSession object for a session upon initiation of said session and updates said
3 HttpSession object for said session as said session progress.

1 Claim 20 (previously presented): The system of claim 19 further comprising a plurality of
2 Java Virtual Machines (JVMs) of which different ones of said JVMs may service different
3 http requests corresponding to a single session and wherein said memory is accessible to each
4 of said JVMs.

1 Claim 21 (currently amended): The system of claim 20 wherein said first program stores said
2 HttpSession object for each session handled by a JVM in a memory local to said JVM and
3 wherein said second program writes a copy of said HttpSession object for each http session
4 stored in said ~~local memories to~~ said database local to said database server.

1 Claim 22 (original): The system of claim 21 wherein said at least one network server
2 comprises a plurality of network servers and wherein different ones of said JVMs run on
3 different ones of said network servers.

1 Claim 23 (original): The system of claim 22 wherein said server system services the World
2 Wide Web.

1 Claim 24 (original): The system of claim 16 wherein said particular time is a function of a
2 periodic interval.

1 Claim 25 (original): The system of claim 24 wherein said periodic interval is a day and said
2 particular time is a time of day.

1 Claim 26 (original): The system of claim 25 wherein said time of day is a time of day that
2 network traffic involving said server system is expected to be low.

1 Claim 27 (original): The system of claim 16 further comprising:
2 a computer program for determining a volume of network traffic involving said server
3 system; and
4 wherein said particular time is a function of said network traffic involving said server
5 system.

1 Claim 28 (original): The system of claim 27 wherein said particular time is further a function
2 of a predetermined interval.

1 Claim 29 (currently amended): The system of claim 16 wherein said ~~third~~ forth program
2 invalidates all of said ~~sessions~~ copies of said session data stored in said database at said
3 particular time.

1 Claim 30 (currently amended): The system of claim 16 wherein, for each copy of session
2 data in said database, said ~~third~~ forth program determines if said session has timed out and
3 invalidates the copy of session data corresponding to said sessions that have been determined
4 to have timed out.

1 Claim 31 (currently amended): A method of maintaining HttpSession objects in a server
2 system serving a network, said server system including a plurality of network servers running
3 a plurality of Java Virtual Machines (JVMs), said method comprising the steps of:

4 (1) storing in a memory local to a network server running a JVMs HttpSession objects
5 for each session serviced by said JVMs;

6 ~~{{(1)}}~~ (2) storing in a database accessible to all of said JVMs copies of said
7 HttpSession objects for each session serviced by said JVMs;

8 ~~{{(2)}}~~ (3) performing contemporaneous time out testing of a particular HttpSession
9 object every time a request is received for said particular HttpSession object prior to utilizing
10 said_particular HttpSession object, and not invalidating a copy of said particular HttpSession
11 object in said database even if said contemporaneous testing has indicated that the
12 corresponding session has timed out; and

13 ~~{{(3)}}~~ (4) performing an invalidation procedure on said copies of said HttpSession
14 objects at a particular time that is independent of when said contemporaneous testing is
15 performed.

1 Claim 32 (original): The method of claim 31 wherein said HttpSession objects conform to
2 the Java servlet APIs are J2EE servlet APIs.

1 Claim 33 (currently amended): The method of claim 32 wherein step (1) comprises the steps
2 of:

3 (1.1) creating ~~[[an]]~~ said HttpSession object for a session upon initiation of said
4 session and storing said HttpSession object in a memory local to a particular one of said
5 JVMs upon initiation of said session;

6 ~~(1.2) writing a copy of said HttpSession object for each session stored in said local~~
7 ~~memory to said database upon said creation;~~

8 ~~(1.3)~~ (1.2) updating said HttpSession object for each said http session in said local
9 memory as said session progresses.

10 and wherein step (2) comprises the steps of:

11 (2.1) writing a copy of said HttpSession object for each session stored in said local
12 memory to said database upon said creation;

13 ~~(1.3)~~ (2.2) updating said copy of said corresponding HttpSession object in said
14 database as said session progresses.

1 Claim 34 (original): The method of claim 32 wherein said particular time is a function of a
2 periodic interval.

1 Claim 35 (original): The method of claim 34 wherein said periodic interval is a day and said
2 particular time is a time of day when network traffic involving said server system is expected
3 to be low.

1 Claim 36 (currently amended) The method of claim 31 further comprising the steps of:
2 ~~(4)~~ (5) determining a volume of network traffic involving said server system; and
3 wherein said particular time is a function of said network traffic involving said server
4 system.

1 Claim 37 (currently amended) The method of claim 31 wherein said invalidation procedure
2 comprises invalidating all of said ~~sessions~~ copies of said HttpSession objects stored in said
3 database at said particular time.

1 Claim 38 (currently amended) The method of claim 31 wherein said invalidation procedure
2 comprises the steps of:
3 (3.1) for each copy of an HttpSession object in said database, determining if said
4 corresponding session has timed out; and
5 (3.2) invalidating each copy of an HttpSession object in said database that has timed
6 out.